

**Notice of Allowability**

Application No.

09/627,413

Applicant(s)

TOCK ET AL.

Examiner

Art Unit

Tuan A Vu

2124

**-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address--**

All claims being allowable, PROSECUTION ON THE MERITS IS (OR REMAINS) CLOSED in this application. If not included herewith (or previously mailed), a Notice of Allowance (PTOL-85) or other appropriate communication will be mailed in due course. **THIS NOTICE OF ALLOWABILITY IS NOT A GRANT OF PATENT RIGHTS.** This application is subject to withdrawal from issue at the initiative of the Office or upon petition by the applicant. See 37 CFR 1.313 and MPEP 1308.

1. ☒ This communication is responsive to 10/29/2004.
2. ☒ The allowed claim(s) is/are 22-24, 26-30, 32-34, 36-40, 52-55, 57-60, and 62-65.
3. ☒ The drawings filed on 27 July 2000 are accepted by the Examiner.
4. ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
  - a) ☐ All    b) ☐ Some\*    c) ☐ None    of the:
  1. ☐ Certified copies of the priority documents have been received.
  2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
  3. ☐ Copies of the certified copies of the priority documents have been received in this national stage application from the International Bureau (PCT Rule 17.2(a)).

\* Certified copies not received: \_\_\_\_\_.

Applicant has THREE MONTHS FROM THE "MAILING DATE" of this communication to file a reply complying with the requirements noted below. Failure to timely comply will result in ABANDONMENT of this application.  
**THIS THREE-MONTH PERIOD IS NOT EXTENDABLE.**

5. ☐ A SUBSTITUTE OATH OR DECLARATION must be submitted. Note the attached EXAMINER'S AMENDMENT or NOTICE OF INFORMAL PATENT APPLICATION (PTO-152) which gives reason(s) why the oath or declaration is deficient.
  6. ☐ CORRECTED DRAWINGS (as "replacement sheets") must be submitted.
    - (a) ☐ including changes required by the Notice of Draftsperson's Patent Drawing Review (PTO-948) attached
      - 1) ☐ hereto or 2) ☐ to Paper No./Mail Date \_\_\_\_\_.
    - (b) ☐ including changes required by the attached Examiner's Amendment / Comment or in the Office action of Paper No./Mail Date \_\_\_\_\_.
- Identifying Indicia such as the application number (see 37 CFR 1.84(c)) should be written on the drawings in the front (not the back) of each sheet. Replacement sheet(s) should be labeled as such in the header according to 37 CFR 1.121(d).
7. ☐ DEPOSIT OF and/or INFORMATION about the deposit of BIOLOGICAL MATERIAL must be submitted. Note the attached Examiner's comment regarding REQUIREMENT FOR THE DEPOSIT OF BIOLOGICAL MATERIAL.

**Attachment(s)**

1. ☒ Notice of References Cited (PTO-892)
2. ☐ Notice of Draftperson's Patent Drawing Review (PTO-948)
3. ☒ Information Disclosure Statements (PTO-1449 or PTO/SB/08),  
Paper No./Mail Date 20000727
4. ☐ Examiner's Comment Regarding Requirement for Deposit  
of Biological Material
5. ☐ Notice of Informal Patent Application (PTO-152)
6. ☐ Interview Summary (PTO-413),  
Paper No./Mail Date \_\_\_\_\_.
7. ☒ Examiner's Amendment/Comment
8. ☒ Examiner's Statement of Reasons for Allowance
9. ☐ Other \_\_\_\_\_.

### DETAILED ACTION

1. This action is responsive to the Applicant's response filed 10/29/2004.

As indicated in Applicant's response, claims 22, 24, 26-30, 32, 34, 36-40 have been amended and claims 52-61 added. Claims 22-24, 26-30, 32-34, 36-40, and 52-61 are pending in the office action.

### EXAMINER'S AMENDMENT

2. An examiner's amendment to the record appears below. Should the changes and/or additions be unacceptable to applicant, an amendment may be filed as provided by 37 CFR 1.312. To ensure consideration of such an amendment, it MUST be submitted no later than the payment of the issue fee.

Authorization for this examiner's amendment was given in a telephone interview with Leonard Heyman, Reg # 40,418 on 01/05/2005.

The application has been amended as follows:

#### Claim 22:

A method of generating object-oriented computer programs for accessing and updating ~~persistently stored objects~~ of an object-oriented database stored in persistent storage, the method comprising:

receiving an initial computer program that includes original instructions for accessing objects stored in a ~~computer's main memory~~ of a computer, the original instructions including object accessing instructions for accessing ~~persistent~~ the objects comprising main memory copies of persistently stored objects;

scanning the initial computer program to automatically identify said object accessing instructions and corresponding program locations ~~representing a first set of identified program locations~~;

Art Unit: 2124

automatically, under computer program control, revising the initial computer program to generate a revised computer program, said revising comprising adding a load data method call ~~modifying the persistent objects, and said revising further comprising adding object loading instructions~~ to the initial computer program at the ~~first set of the identified~~ program locations, wherein the added ~~object loading instructions~~ load data method call, during execution of the revised computer program, load calls a load data method, the load data method call causing data from ~~respective ones of the persistent~~ objects from the object-oriented database in the persistent storage of the computer to be loaded into persistent objects in the main memory when each the ~~respective~~ object is accessed and the respective object is not already in the main memory; and

wherein each persistent object of the persistent objects includes a persistent data descriptor created by the load data instructions; and wherein said modifying the persistent objects comprises modifying object data structures of said persistent objects to store persistent data descriptors, said modifying revising further comprising adding new methods to the persistent objects, said new methods comprising code one or more predefined object classes allowing access and use of said persistent data descriptors descriptor, the load data method being a method of the one or more predefined object classes.

**Claim 23:**

The method of claim 22, wherein the added ~~object loading instructions~~ load data method ~~is~~ are inactive during execution of the revised computer program except when ~~[[a]]~~ the respective object to be accessed is referenced by a null location indicator indicating the respective object has not been loaded into a corresponding persistent object in the main memory.

**Claim 24:**

The method of claim 22, wherein:

~~each of said persistent data descriptor descriptors includes a pointer to a next dirty object;~~  
the one or more predefined object classes include a mark-object-as-dirty method and an end-transaction method wherein the mark-object-as-dirty method causes ~~said revising further includes adding code that adds~~ objects containing new and/or updated data to be added to a linked list of dirty objects using said the pointer to said the next dirty object; and object and the end-transaction method causes the ~~said revising further includes adding object storing~~

Art Unit: 2124

~~instructions to the initial computer program that, during execution of the revised computer program, store said objects in said linked list of dirty objects~~ to be stored into the persistent storage.

**Claim 26:**

A method of generating object-oriented computer programs for accessing and updating ~~persistently stored objects of an object-oriented database stored in persistent storage~~, the method comprising:

receiving an initial computer program that includes original instructions for accessing and updating objects stored in ~~a computer's main memory of a computer~~, and for committing transactions in which one or more objects may have been updated, the original instructions including instructions for accessing ~~persistent the~~ objects comprising main memory copies of persistently stored objects;

scanning the initial computer program to automatically identify object updating instructions and transaction commit instructions;

automatically, under computer program control, revising the initial computer program to generate a revised computer program ~~by:~~ by adding load data instructions to the initial computer program, the load data instructions, during execution of the revised computer program, load data from respective ones of the objects in the object-oriented database in the persistent storage into modifying the persistent objects in main memory, wherein ~~said modifying the persistent objects comprising modifying object data structures of said persistent objects to store~~ comprise persistent data descriptors created by the load data instructions, said modifying the persistent objects further comprises adding new methods to the persistent objects, the new methods comprising code allowing access and use of said persistent data descriptors;

wherein the new code includes adding dirty object marking instructions to the initial computer program that, during execution of the revised computer program, modifies the persistent data descriptors of dirty ones the persistent objects that ~~dirty objects which~~ contain new and/or updated data so that ~~said dirty~~ the dirty ones of the persistent objects can be identified; and

wherein the new code further includes adding object storing end transaction instructions to the initial computer program that, during execution of the revised computer program, store

Art Unit: 2124

said dirty ones of the persistent objects ~~in the computer's main memory~~ into the persistent storage, wherein the persistent data descriptors ~~stored in said object data structures~~ of the persistent object are used by the ~~object storing~~ end transaction instructions to identify the ~~certain~~ respective dirty ones of the persistent objects.

**Claim 27:**

The method of claim 26, wherein the persistent data descriptors includes a persistent storage object identifier, the ~~object storing instructions~~ new code including instructions for replacing local object references in the ~~certain~~ respective objects with the persistent storage object identifiers in corresponding ones of the data descriptors before storing the dirty ones of the persistent ~~certain~~ respective objects in the persistent storage, wherein the local object references reference objects in the main memory and the persistent storage object identifiers reference objects in the persistent storage.

**Claim 28:**

A method of generating object-oriented computer programs for accessing and updating ~~persistently stored~~ objects of an object-oriented database stored in persistent storage, the method comprising:

scanning an initial computer program to automatically identify object accessing instructions and object updating instructions, the initial computer program including original instructions for accessing and updating the objects as though the objects were stored in a computer's main memory, the ~~original~~ object accessing and object updating instructions including instructions for accessing ~~persistent~~ and updating the objects comprising main memory copies of persistently stored objects;

automatically revising the initial computer program to generate a revised computer program by ~~modifying the persistent objects and adding supplemental~~ load data instructions to the initial computer program, the load data instructions loading data from respective objects from the object-oriented database in the persistent storage into ~~wherein said modifying the persistent objects in the main memory, wherein the persistent objects each include~~ comprises modifying object data structures of the persistent objects to store persistent data descriptors therein, said modifying the persistent objects the revising further comprising adding new ~~methods to the persistent objects, the new methods comprising~~ code allowing access and use of the persistent

Art Unit: 2124

data descriptors, wherein each of said persistent data descriptors including includes a pointer to a next dirty object, said supplemental instructions the new code including mark object instructions that, when a persistent object becomes dirty by containing new and/or updated data, adding add the dirty persistent objects object to a linked list of dirty objects using said pointer to a next dirty object in said persistent data descriptors, the dirty objects being ones of the persistent objects that when said object contains contain new and/or updated data.

**Claim 29:**

The computer implemented method of claim 28, wherein ~~said supplemental instructions further includes instructions loading~~ the load data instructions load the respective ones of the objects from the persistent storage of the computer into the main memory of the computer when each of the respective ~~object is~~ objects are accessed and ~~the respective object is~~ not already in the main memory.

**Claim 30:**

The computer implemented method of claim 29, wherein ~~said supplemental instructions the additional code~~ further includes end transaction instructions for storing objects in the main memory that are members of the linked list of dirty objects.

**Claim 32:**

A computer program product for use in a conjunction with a computer having a main memory ~~and persistent storage~~, the computer program product comprising a computer readable storage medium and a computer program mechanism embedded therein, the computer program mechanism comprising:

a postprocessor procedure for modifying an initial computer program that includes original instructions for accessing and updating objects stored in a ~~computer's main memory~~ persistent storage in an object-oriented database, the original instructions including instructions for accessing ~~persistent~~ the objects comprising main memory copies of persistently stored objects;

the postprocessor procedure including instructions for:

receiving ~~an~~ the initial computer program ~~that includes original instructions for~~ accessing objects ~~stored in a computer's main memory~~;

Art Unit: 2124

scanning the initial computer program to automatically identify object accessing instructions and corresponding program locations ~~representing a first set of identified program locations;~~

automatically, under computer program control, revising the initial computer program to generate a revised computer program, said revising comprising adding a load data method call ~~modifying the persistent objects, and said revising further comprising adding object loading instructions~~ to the initial computer program at the ~~first set of the identified~~ program locations, wherein the added ~~object loading~~ load data method call instructions, during execution of the revised computer program, ~~load~~ calls a load data method, the load data method call causing data from respective ~~ones of the persistent~~ objects from the object-oriented database in the persistent storage of the computer into persistent objects in the main memory when each ~~the~~ respective object is accessed and the respective object is not already in the main memory; and

wherein each persistent object of the persistent object includes a persistent data descriptor created by the load data instructions; ~~said modifying the persistent objects comprises modifying object data structures of said persistent objects to store persistent data descriptors, said modifying the revising~~ further comprising adding new methods to ~~the persistent objects, said new methods comprising code~~ one or more predefined object classes allowing access and use of said persistent data descriptors descriptor, the load data method being a method of the one or more predefined object classes.

**Claim 33:**

The computer program product of claim 32, wherein the added ~~object loading instructions~~ load data method ~~is~~ are inactive during execution of the revised computer program except when ~~[[a]]~~ the respective object to be accessed is referenced by a null location indicator, indicating the respective object has not been loaded into a corresponding persistent object in the main memory.

Art Unit: 2124

**Claim 34:**

The computer program product of claim 32, wherein:

~~each of said persistent data descriptors~~ descriptor includes a pointer to a next dirty object;  
the one or more predefined object classes comprise a mark-object-as-dirty method and an end-transaction method wherein the mark-object-as-dirty method causes said revising further includes adding code that adds objects containing new and/or updated data to be added to a linked list of dirty objects using said the pointer to said the next dirty object; and object and the end-transaction method causes the said revising further includes adding object storing instructions to the initial computer program that, during execution of the revised computer program, store said objects in said linked list of dirty objects to be stored into the persistent storage.

**Claim 36:**

A computer program product for use in conjunction with a computer having a main memory ~~and persistent storage~~, the computer program product comprising a computer readable storage medium and a computer program mechanism embedded therein, the computer program mechanism comprising:

a postprocessor procedure for modifying an initial computer program that includes original instructions for accessing and updating objects stored in a ~~computer's main memory~~ persistent storage in an object-oriented database, the original instructions including instructions for accessing ~~persistent~~ the objects comprising main memory copies of persistently stored objects;

the postprocessor procedure including instructions for:

receiving ~~an the~~ initial computer program ~~that includes original instructions for accessing and updating objects stored in a computer's main memory and for committing transactions in which one or more objects may have been updated;~~

scanning the initial computer program to automatically identify object updating instructions and transaction commit instructions;

automatically, under computer program control, revising the initial computer program to generate a revised computer program ~~by~~ by adding load data instructions to the initial computer program, the load data instructions, during execution of the revised computer



Art Unit: 2124

program, load data from respective ones of the objects in the object-oriented database in the persistent storage into modifying the persistent objects in main memory, said modifying the persistent objects comprising modifying object data structures of said persistent objects to store persistent data descriptors created by the load data instructions, said modifying the persistent objects further comprises adding new methods to the persistent objects, the new methods comprising code allowing access and use of said persistent data descriptors;

wherein the new code includes adding dirty object marking instructions to the initial computer program that, during execution of the revised computer program, modifies the persistent data descriptors of dirty ones of the persistent objects that ~~dirty objects~~ contain new and/or updated data so that ~~said dirty~~ the dirty ones of the persistent objects can be identified; and

wherein the new code further includes adding object storing end transaction instructions to the initial computer program that, during execution of the revised computer program, store said dirty ones of the persistent objects ~~in the computer's main memory~~ into the persistent storage, wherein the persistent data descriptors ~~stored in said object data structures~~ of the persistent object are used by the ~~object storing~~ end transaction instructions to identify the ~~certain~~ respective dirty ones of the persistent objects.

**Claim 37:**

The computer program product of claim 36, wherein the persistent data descriptors includes a persistent storage object identifier, the ~~object storing instructions~~ new code including instructions for replacing local object references in the certain respective objects with the persistent storage object identifiers in corresponding ones of the data descriptors before storing the dirty ones of the persistent ~~certain respective~~ objects in the persistent storage, wherein the local object references reference objects in the main memory and the persistent storage object identifiers reference objects in the persistent storage.

**Claim 38:**

A computer program product for use in conjunction with a computer having a main memory ~~and persistent storage~~, the computer program product comprising a computer readable storage medium and a computer program mechanism embedded therein, the computer program mechanism comprising:

Art Unit: 2124

a procedure for modifying an initial computer program that includes original instructions for accessing and updating objects stored in a ~~computer's main memory~~ persistent storage in an object-oriented database, the original instructions including instructions for accessing ~~persistent~~ the objects comprising main memory copies of persistently stored objects;

the procedure including instructions for:

scanning an initial computer program to automatically identify ~~object-accessing~~ the instructions for accessing and instructions for and-object updating instructions;

automatically revising the initial computer program to generate a revised computer program by ~~modifying the persistent objects and adding supplemental~~ adding load data instructions to the initial computer program, the load data instructions loading data from respective objects from the object-oriented database in the persistent storage into wherein said ~~modifying the persistent objects in the main memory, wherein the persistent objects each include~~ comprises modifying object data structures of the persistent objects to store persistent data descriptors therein, said modifying the persistent objects the revising further comprising adding new methods to the persistent objects, the new methods comprising code allowing access and use of the persistent data descriptors, wherein each of the persistent data descriptors ~~including~~ includes a pointer to a next dirty object, ~~said supplemental instructions~~ the new code including mark object instructions that, when a persistent object becomes dirty by containing new and/or updated data, add the dirty adding persistent objects object to a linked list of dirty objects using the pointer to a next dirty object in the persistent data descriptors, the dirty objects being ones of the persistent objects that when the object contains contain new and/or updated data.

**Claim 39:**

The computer program product of claim 38, wherein the ~~supplemental instructions further~~ includes instructions loading load data instructions load the respective ones of the objects from the persistent storage of the computer into the main memory of the computer when each of the respective ~~object is~~ objects are accessed and ~~the respective object is~~ not already in the main memory.

**Claim 40:**

Art Unit: 2124

The computer program product of claim 39, wherein ~~said supplemental instructions~~ the additional code further includes end transaction instructions for storing objects in the main memory that are members of the linked list of dirty objects.

**Claim 52:**

The method of claim 22 wherein each ~~of said persistent data descriptors~~ descriptor comprises a full DBMS database object identifier.

**Claim 53:**

The method of claim 22 wherein ~~each of said persistent data descriptors~~ descriptor is referenced by a corresponding pointer in a corresponding one of ~~said~~ the respective objects ~~object data structures~~.

**Claim 54:**

The method of claim 24 wherein ~~said code that adds objects containing new and/or updated data to a linked list of dirty objects~~ mark-object-as-dirty method comprises code to copy an object pointer in a list header into said pointer to a next dirty object and code to store a pointer to said object containing new and/or updated data in said list header.

**Claim 55:**

The method of claim 22 wherein each ~~of said persistent data descriptors~~ descriptor includes a full object identifier of each reference in the persistent object to another object.

**Claim 56.** (Canceled)

**Claim 57:**

The computer program product of claim 32 wherein ~~each of said persistent data descriptors~~ descriptor comprises a full DBMS database object identifier.

**Claim 58:**

Art Unit: 2124

The computer program product of claim 32 wherein ~~each of~~ said persistent data ~~descriptors~~ descriptor is referenced by a corresponding pointer in a corresponding one of said object data structures.

**Claim 59:**

The computer program product of claim 34 wherein said ~~code that adds objects containing new and/or updated data to a linked list of dirty objects~~ mark-object-as-dirty method comprises code that, for each of said objects containing new and/or updated data, to copy copies an object pointer in a list header into said pointer to a next dirty object and code to store a pointer to said each object containing new and/or updated data in said list header.

**Claim 60:**

The computer program product of claim 22 wherein ~~each of~~ said persistent data ~~descriptors~~ descriptor includes a full object identifier of each reference in the persistent object to another object.

**Claim 61.** (Canceled)

**Claim 62.** (New) The method of claim 26 wherein the load data instructions comprise a call to a load data method in an object class used by the revised program.

**Claim 63.** (New) The method of claim 62 wherein the new code comprises a database object class, the load data method being a method of the database object class.

**Claim 64.** (New) The method of claim 26 wherein the load data instructions comprise a call to a load data method, the dirty object marking instructions comprise a call to a dirty object marking method, and the end transaction instructions comprise a call to an end transaction method.

Art Unit: 2124

**Claim 65.** (New) The method of claim 64 wherein the new code further comprises a database object class wherein the load data method, the dirty object marking method, and the end transaction method are methods of the database object class.

***EXAMINER'S STATEMENT OF REASONS FOR ALLOWANCE***

3. In view of the above examiner's amendment including new claims, Claims 22-24, 26-30, 32-34, 36-40, 52-55, 57-60, and 62-65 are pending in the office action.

Claims 22-24, 26-30, 32-34, 36-40, 52-55, 57-60, and 62-65 are allowed.

The following is an examiner's statement of reasons for allowance.

The prior art of record, taken alone or in combination fails to teach or suggest the following claimed features:

A method for accessing and updating persistently stored object, the method comprising (i) scanning a initial computer program including instructions for accessing objects being main memory copies of persistently stored objects to identify corresponding program locations of said instructions for accessing persistently stored database objects; and (ii) automatically revising the program to add a load data method call for causing respective ones of the persistent objects to load from a database into the main memory when each respective object is not already in said memory when being accessed, and (iii) said persistent objects including a persistent data descriptor and said modifying further adding new method calls to one or more predefined object classes allowing access and use of said persistent data descriptor, the load data method being one such plurality of methods --- as recited in claim 22 and 32.

A method as above with scanning as in (i) comprising adding a load instructions to cause persistent objects to load as in (ii) and add new code for modifying descriptors as in (iii); said

Art Unit: 2124

revising further comprising either (iv) a mark dirty method to mark said objects as dirty persistent objects when they are found old or updated; and to invoke a end transaction method to persist those marked objects to persistent storage using the persistent object descriptors -- as recited in claims 26 and 36; or (v) the added code including said dirty marking method that further comprises adding said marked dirty objects to a linked list using a pointer in said descriptors --- as recited in claims 28 and 38.

**Hastings**, USPN: 5,193,180, discloses a error-checking code expansion method with identifying of locations in a program at binding time to establish code insertion points in conjunction with predefined offset/index information and symbol/relocation table; such insertion to insert additional instructions for monitoring memory access status or allocation potential errors or bounds violation; but does not disclose identifying instructions for accessing persistent objects related to a Object Oriented database as in (i) and to add to the location thus identified a load method call causing persistent objects to be loaded to memory if it is not already loaded as in main memory as in (ii) and further causing modifying persistent data descriptors of said persistent objects and adding plurality of instructions allowing access and use of said descriptors as in (iii) or (iv) or (v)

**Gupta**, USPN: 5,822,590, in a method to maintain heap memory, discloses providing a pointer structure referring to an identifier of a persistent object, and mapping tool to dynamically deference such pointer information for memory runtime access thereby not increasing heap size; but does not teach scanning a program to identify persistent object accessing instructions as in (i); and adding a load method call to perform the loading as in (ii) with additional added instructions to modify/use persistent object descriptors as in (iii) or (iv) or (v).

Many prior art (e.g. **Levine**, USPN: 6134710) teach about load instructions being inserted by the compiler, to prevent cache miss and/or alleviate memory access time or table walks; and this prefetch or preload methodology does not teach about modifying the initial program at specific locations where an persistent/database stored object access instruction is identified; and invoking at which locations a load method call under program control for accessing said persistent objects and load respective persistent objects to memory for the first time as in (ii) and causing persistent objects to be modify and use the persisted objects internal descriptors, such load method call further used for adding more modifying and access instructions as in (iii) or (iv) or (v).

Any comments considered necessary by applicant must be submitted no later than the payment of the issue fee and, to avoid processing delays, should preferably accompany the issue fee. Such submissions should be clearly labeled "Comments on Statement of Reasons for Allowance."

### ***Conclusion***

4. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Tuan A Vu whose telephone number is (272) 272-3735. The examiner can normally be reached on 8AM-4:30PM/Mon-Fri.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kakali Chaki can be reached on (571)272-3719.

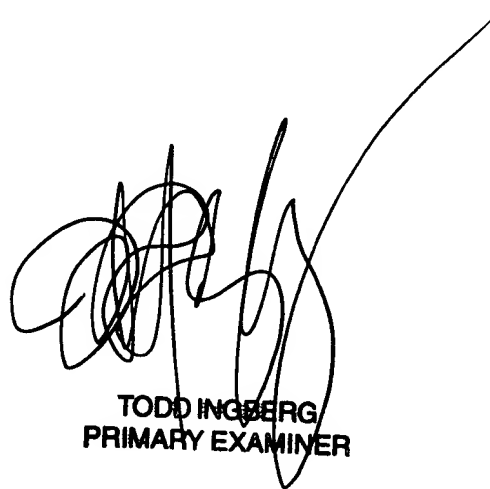
The fax phone number for the organization where this application or proceeding is assigned is (571) 273-3735 ( for non-official correspondence – please consult Examiner before

Art Unit: 2124

using) or 703-872-9306 ( for official correspondence) or redirected to customer service at 571-272-3609.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

VAT  
January 5, 2005



TODD INGERBERG  
PRIMARY EXAMINER